

Built to Teach

Learning science in a voice tutor: the design principles of DuckyHelper

The DuckyHelper team, MingLLM, Inc. October 2026 • Version 1.0 • hello@dukyhelper.com

Abstract

DuckyHelper is a voice tutor for learners of any age and subject. It runs as a Mac app that can see and draw on the learner's screen, and as a web app, Eddy, that teaches on its own desk. This white paper describes how the tutor is designed from published learning science. We take ten findings, from Bloom's two sigma result to the risk of unguarded AI help reported by Bastani et al. (2025), and for each one describe the teaching rules and the interface tools that put it into practice. The tutor is trained on 1,000,000+ tutoring examples, and its design is inspired by 100+ learning-science papers; this paper cites the ones each design choice rests on. We then describe where the principles pull against each other and against a fast product. DuckyHelper has not been evaluated in a controlled study. This paper reports no learning outcomes, no usage numbers and no benchmark scores of our own. Every number in it belongs to the study that reported it.

Keywords: AI tutoring, learning science, intelligent tutoring systems, retrieval practice, spaced repetition, worked examples, multimedia learning, scaffolding

1 Introduction

Getting an answer has never been easier. Learning how to get it has not changed. In a field experiment with high school math students, Bastani et al. (2025) found that a plain GPT-4 chat raised grades during practice and lowered them on the exam once it was taken away, while a version built to protect learning largely avoided the harm:

“unfettered access to GPT-4 can harm educational outcomes”

(Bastani et al., 2025)

The opposite case is old. Bloom (1984) reported that students taught one on one by a tutor did about two standard deviations better than students in a regular class, and that such tutoring was

“too costly for most societies to bear on a large scale”

(Bloom, 1984)

DuckyHelper is our attempt at a tutor for anyone that is a tutor, not an answer machine: **built to teach**, and **trained to teach**. This paper explains what that means in the design. Section 2 describes the tutor. Section 3 takes ten principles from learning science and, for each, the evidence and what the tutor does about it. Section 4 covers the tensions between them. Section 5 lists what this paper does not show.

2 The tutor

2.1 Two surfaces, one tutor

In the **Mac app**, the learner presses fn and talks. A small pill opens at the bottom of the screen. While it is open, the tutor, called Ducky there, keeps a small current view of the screen, so it can answer about the exact thing the learner means. It can point at and draw on the learner's real screen over any app, walk them through steps in their own apps while watching their clicks and typing, and open cards in the pill: quizzes, hints, worked steps, graphs, simulations and flashcards.

In the **web app** at app.duckyhelper.com, the tutor is called Eddy. It never sees the learner's screen. It teaches on its own desk: a blackboard that it draws on in time with its voice, pictures it labels part by part, quizzes and flashcards beside them, and a reader for articles and books. Both share one account, one memory of the learner and one set of flashcards.

2.2 Three layers

We describe the design in three layers. The **model** is trained on 1,000,000+ tutoring examples. The **teaching rules** are written instructions the tutor follows in every session; this paper describes them in our own words. The **tools** are the pieces of interface the tutor can open for the learner, named here as the learner sees them. A principle usually shows up in all three.

IN DUCKYHELPER

The tutor starts from what the learner thinks: a question first, then a hint or a guiding question, and the learner takes the next step. It praises effort and the specific good moves the learner makes.

3 Ten principles and what the tutor does

Table 1. Each principle, the core evidence and the behaviour it drives. Mac = Mac app only.

Principle	Core evidence	What the tutor does
Teach one on one	(Bloom, 1984; VanLehn, 2011)	A live voice tutor for one learner; asks before it explains
Guide, don't hand over answers	(Bastani et al., 2025; Macina et al., 2023a)	Try first; hints before answers
Scaffold, then fade	(Wood et al., 1976; Kalyuga et al., 2003)	Hint ladder (Mac); tutorials that wait for each step (Mac)
Worked examples first	(Sweller & Cooper, 1985; Atkinson et al., 2000)	Worked steps, one move each; similar examples
Pictures with words	(Mayer & Moreno, 2003; Fiorella & Mayer, 2016)	Draws on the screen (Mac) and on the blackboard as it talks
Retrieval practice	(Roediger & Karpicke, 2006; Karpicke & Roediger, 2008)	A check question after each step; quizzes
Spaced practice	(Cepeda et al., 2006; Dunlosky et al., 2013)	Flashcards on a spaced schedule; misses become cards
Interleaving	(Rohrer & Taylor, 2007; Kornell & Bjork, 2008)	Reviews mix due cards across topics; mixed quizzes on request
Self-explanation	(Chi et al., 1994a; Chi & Wylie, 2014)	Asks what you think; writes your spoken steps on the whiteboard (Mac)
Feedback on the task	(Hattie & Timperley, 2007; Kluger & DeNisi, 1996)	Not quite and why; the first wrong line; a mark on the exact step

3.1 Teach one on one

Evidence. Bloom's result has been revisited many times and the effect is smaller than two standard deviations in larger reviews, but it is real. VanLehn (2011) found human tutoring at an effect size of 0.79 and step-based computer tutors close to it, and Nickow et al. (2024) found a pooled effect of 0.288 standard deviations across randomized tutoring programs for children. How tutors interact matters. In Chi et al. (2001), tutors were told to stop explaining and only prompt, and their students learned as well:

“students learned just as effectively even when tutors were suppressed from giving explanations and feedback”

(Chi et al., 2001)

Design. The tutor talks with one learner at a time, out loud, and its rules make every lesson a dialogue. It infers the learner's level from their words and their page, speaks their language, and remembers how they learn between sessions (Memory) and which slips they make (Mistakes).

IN DUCKYHELPER

Every lesson goes back and forth. Before a big topic the tutor asks what the learner already knows, then explains in short steps and asks a quick question after each one, waiting for the answer before moving on.

3.2 Guide, don't hand over answers

Evidence. In Bastani et al. (2025), grades during practice rose 48% with the plain chat and 127% with the guarded tutor; on the exam without AI, the plain chat group did 17% worse than students who never had access. The authors wrote:

“These negative learning effects are largely mitigated by the safeguards in GPT Tutor”
(Bastani et al., 2025)

Language models are good solvers and poor teachers by default; Macina et al. (2023a) found they reveal solutions too early. In older tutoring systems, students who used hints only to reach answers learned much less (Baker et al., 2004).

Design. The rules separate questions of fact, which get a short answer and one small next step, from problems the learner is learning to solve:

IN DUCKYHELPER

On homework and graded work the tutor does not hand over the final answer. The learner tries first and is then guided one step at a time. A learner who is truly stuck works through a similar problem with the tutor, then goes back to their own.

Our companion white paper, *Hints before answers*, covers this principle in detail.

3.3 Scaffold, then fade

Evidence. Wood et al. (1976) described the tutor taking over the parts of a task beyond the learner, so the learner can finish the rest. Vygotsky (1978) located useful help in

“the distance between the actual developmental level as determined by independent problem solving”
(Vygotsky, 1978)

and the level reached with guidance. Strong guidance helps beginners (Kirschner et al., 2006), but the same help can stop working as skill grows (Kalyuga et al., 2003).

Design. In the Mac app, a learner stuck on a problem gets a hint ladder they open one rung at a time: a nudge, a bigger hint, the worked step, and the answer last. Hints used count against the tutor's estimate of their mastery of that skill. Step-by-step tutorials in the learner's own apps move on only when the learner has really done the step, and “watch me, then you” demonstrates once and then hands over control.

3.4 Worked examples first

Evidence. For beginners, studying solved problems beats solving them unaided. Sweller and Cooper (1985) found that

“subsequent problems similar to the initial ones also were solved more rapidly”
(Sweller & Cooper, 1985)

and Atkinson et al. (2000) advise placing examples next to matched practice. Renkl (2014) calls examples a very effective means of *initial* skill acquisition.

Design. Worked steps come as 3 to 6 steps, one move each, each explaining what changes and why. On homework they stop where the learner is, with a hint for what comes next, and a stuck learner gets a *similar* problem worked with them rather than their own.

IN DUCKYHELPER

On homework, worked steps stop where the learner is and add a hint for the next step, instead of showing the whole solution.

3.5 Pictures with words

Evidence. Mayer and Moreno (2003) summarize multimedia learning as building connections between pictures and words, and list ways to keep either channel from overloading. Watching an instructor draw beat seeing finished diagrams for learners with little prior knowledge (Fiorella & Mayer, 2016).

Design. In the Mac app the tutor draws on the learner's own screen, one mark at a time, and points whenever it says "this". On Eddy's blackboard, speech and drawing are locked together:

IN DUCKYHELPER

On the blackboard, each sentence is heard just as its drawing appears, so speech and picture stay in step.

Our companion design note, *Showing, not telling*, covers the screen pen in detail.

3.6 Retrieval practice

Evidence. Testing is a way to learn, not just to measure. In Roediger and Karpicke (2006), students who took a practice test recalled 56% of a passage after a week, against 42% for students who re-read it, and in Karpicke and Roediger (2008)

"Repeated studying after learning had no effect on delayed recall, but repeated testing produced a large positive effect"

(Karpicke & Roediger, 2008)

Design. Check questions are built into how the tutor talks: one after each step, one after explaining a page, and a recap with a question at the end of every lesson. Quizzes open in the moment with multiple choice, true or false, typed math, blanks, ordering, matching and labelling.

IN DUCKYHELPER

Quick spoken questions and short quizzes come up throughout a session, not only at the end.

3.7 Spaced practice

Evidence. Across 317 experiments, spaced study beat massed study, and the best gap grew with how long the memory had to last (Cepeda et al., 2006). Dunlosky et al. (2013) rated distributed practice, with practice testing, as the most useful of ten techniques.

Design. Flashcards are scheduled with FSR5, an open spaced repetition algorithm, at its

default parameters and a 90% target recall: each card returns when the learner would likely still remember it about nine times in ten, and the gap grows each time they do. A forgotten card returns within the same review. Missed quiz questions become flashcards by themselves, and when cards are due the tutor offers a short review early in the session.

3.8 Interleaving

Evidence. Mixing problem types in practice helped learners on a later test in Rohrer and Taylor (2007), and mixing examples helped people learn categories in Kornell and Bjork (2008), although it felt less fluent. Dunlosky et al. (2013) rate the evidence as moderate.

Design, and a gap. A review started without choosing a deck pulls every due card across topics, most overdue first, so it mixes subjects. Quizzes are written on request and can be asked for mixed. The tutor does not yet plan interleaved practice by itself.

3.9 Self-explanation

Evidence. Strong students explain worked examples to themselves (Chi et al., 1989), and prompting eighth graders to self-explain a text improved their learning (Chi et al., 1994a). In the ICAP framework, learning rises from passive to active to constructive to interactive engagement (Chi & Wylie, 2014).

Design. The tutor asks what the learner thinks before explaining and asks a question after each step. In the Mac app, when a learner talks through their own steps, the rules have the tutor write each step on a shared whiteboard as it is said and check the math at once. There is no separate “explain it back” mode yet.

3.10 Feedback on the task

Evidence. Feedback is powerful (Hattie & Timperley, 2007) and should be

“nonevaluative, supportive, timely, and specific”

(Shute, 2008)

but it is not automatically good: over 38% of the effects in Kluger and DeNisi (1996) were negative, and

“FI effectiveness decreases as attention moves up the hierarchy closer to the self and away from the task”

(Kluger & DeNisi, 1996)

Design. A wrong quiz answer gets “Not quite” and a one-sentence why; the math check in the Mac app names the first wrong line; the pen can circle the exact step on the learner’s page. The rules keep feedback on the work:

IN DUCKYHELPER

When something is wrong, the tutor first says what went right, then points to the exact step that went wrong.

4 Tensions

4.1 A fast product and effortful learning

We treat speed as a core product value: the tutor should start answering quickly and never make the learner wait on chrome. Learning science warns that the feeling of ease is not learning. Students in active classes learned more and felt they learned less (Deslauriers et al., 2019), and conditions that make practice harder can make it last (Bjork, 1994). We resolve this by being fast at everything except the learner's own thinking: facts are answered at once, the interface responds at once, and the effort we keep is the effort of trying a step.

4.2 Help that fades

The expertise reversal effect (Kalyuga et al., 2003) means a fixed amount of help is wrong for most learners. Rather than guess, the hint ladder lets the learner choose how much help to open, one rung at a time, and the tutor pitches explanations at the level it infers from the conversation.

4.3 Hints as a shortcut

Any hint system can be used to dig for the answer (Baker et al., 2004). The answer sits on the last rung, the rules hold it back until the learner asks twice, and hints used lower the tutor's mastery estimate, which keeps the skill in practice.

IN DUCKYHELPER

The hint ladder is climbed one rung at a time, and the answer is held back until the learner asks for it a second time.

4.4 Being right

A tutor that is wrong teaches the wrong thing, and language models make math errors while tutoring (Miller & DiCerbo, 2024; Daheim et al., 2024). The rules require the tutor to work the math itself before judging the learner, the Mac app checks math line by line with a computer algebra system, and uncertain facts are looked up, with the sources shown to the learner.

IN DUCKYHELPER

Before telling a learner whether their math is right, the tutor works the problem out itself.

5 Limits

- **No outcome evidence yet.** DuckyHelper has not been evaluated in a controlled study. This paper reports no learning outcomes, no usage numbers and no benchmark scores of our own. Every number in it belongs to the study that reported it.
- **Rules are instructions, not guarantees.** The teaching rules are instructions to a language model. A model can fail to follow an instruction, and we have not measured how often this happens.
- **Different learners, different tools.** Most of the research cited here was done with other populations and other tools. That it carries over to a voice tutor is a design bet, not a finding.

- **Some features are Mac only.** Seeing and drawing on the screen, the hint ladder, the math check and the shared whiteboard are in the Mac app; the web app teaches on its own desk.

A fair test of a tutor like this would follow Bastani et al. (2025): randomly assign learners, and measure what they can do later *without* the tool, not how well they do while using it.

6 Conclusion

The research on how people learn is unusually consistent on a few points: learners do best with a guide who makes them think, examples before problems, pictures with words, frequent retrieval, spaced review and feedback on the task. Each is easy to state and easy to lose in an AI that is built to answer. DuckyHelper is built to keep them. Whether it succeeds is a question for a study, and we will publish what it finds.

References

- Atkinson, R. K., Derry, S. J., Renkl, A., & Wortham, D. (2000). Learning from Examples: Instructional Principles from the Worked Examples Research. *Review of Educational Research*, 70(2), 181-214. <https://doi.org/10.3102/00346543070002181>
- Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004). Off-Task Behavior in the Cognitive Tutor Classroom: When Students “Game the System”. *CHI*. <https://doi.org/10.1145/985692.985741>
- Bastani, H., Bastani, O., Sungu, A., Ge, H., Kabakcı, Ö., & Mariman, R. (2025). Generative AI without guardrails can harm learning: Evidence from high school mathematics. *Proceedings of the National Academy of Sciences*, 122(26), e2422633122. <https://doi.org/10.1073/pnas.2422633122>
- Bjork, R. A. (1994). Memory and Metamemory Considerations in the Training of Human Beings. *Metacognition: Knowing about Knowing* (MIT Press, book chapter). <https://doi.org/10.7551/mitpress/4561.003.0011>
- Bloom, B. S. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher*, 13(6), 4-16. <https://doi.org/10.3102/0013189X013006004>
- Cepeda, N. J., Pashler, H., Vul, E., Wixted, J. T., & Rohrer, D. (2006). Distributed practice in verbal recall tasks: A review and quantitative synthesis. *Psychological Bulletin*, 132(3), 354-380. <https://doi.org/10.1037/0033-2909.132.3.354>
- Chi, M. T. H., & Wylie, R. (2014). The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, 49(4), 219-243. <https://doi.org/10.1080/00461520.2014.965823>
- Chi, M. T. H., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-Explanations: How Students Study and Use Examples in Learning to Solve Problems. *Cognitive Science*, 13(2), 145-182. https://doi.org/10.1207/s15516709cog1302_1
- Chi, M. T. H., De Leeuw, N., Chiu, M.-H., & Lavancher, C. (1994a). Eliciting Self-Explanations Improves Understanding. *Cognitive Science*, 18(3), 439-477. https://doi.org/10.1207/s15516709cog1803_3
- Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, 25(4), 471-533. https://doi.org/10.1207/s15516709cog2504_1

- Daheim, N., Macina, J., Kapur, M., Gurevych, I., & Sachan, M. (2024). Stepwise Verification and Remediation of Student Reasoning Errors with Large Language Model Tutors. EMNLP. <https://doi.org/10.18653/v1/2024.emnlp-main.478>
- Deslauriers, L., McCarty, L. S., Miller, K., Callaghan, K., & Kestin, G. (2019). Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences*, 116(39), 19251-19257. <https://doi.org/10.1073/pnas.1821936116>
- Dunlosky, J., Rawson, K. A., Marsh, E. J., Nathan, M. J., & Willingham, D. T. (2013). Improving Students' Learning With Effective Learning Techniques: Promising Directions From Cognitive and Educational Psychology. *Psychological Science in the Public Interest*, 14(1), 4-58. <https://doi.org/10.1177/1529100612453266>
- Fiorella, L., & Mayer, R. E. (2016). Effects of observing the instructor draw diagrams on learning from multimedia messages. *Journal of Educational Psychology*, 108(4), 528-546. <https://doi.org/10.1037/edu0000065>
- Hattie, J., & Timperley, H. (2007). The Power of Feedback. *Review of Educational Research*, 77(1), 81-112. <https://doi.org/10.3102/003465430298487>
- Kalyuga, S., Ayres, P., Chandler, P., & Sweller, J. (2003). The Expertise Reversal Effect. *Educational Psychologist*, 38(1), 23-31. https://doi.org/10.1207/S15326985EP3801_4
- Karpicke, J. D., & Roediger, H. L. (2008). The Critical Importance of Retrieval for Learning. *Science*, 319(5865), 966-968. <https://doi.org/10.1126/science.1152408>
- Kirschner, P. A., Sweller, J., & Clark, R. E. (2006). Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist*, 41(2), 75-86. https://doi.org/10.1207/s15326985ep4102_1
- Kluger, A. N., & DeNisi, A. (1996). The effects of feedback interventions on performance: A historical review, a meta-analysis, and a preliminary feedback intervention theory. *Psychological Bulletin*, 119(2), 254-284. <https://doi.org/10.1037/0033-2909.119.2.254>
- Kornell, N., & Bjork, R. A. (2008). Learning Concepts and Categories: Is Spacing the “Enemy of Induction”? *Psychological Science*, 19(6), 585-592. <https://doi.org/10.1111/j.1467-9280.2008.02127.x>
- Macina, J., Daheim, N., Chowdhury, S., Sinha, T., Kapur, M., Gurevych, I., & Sachan, M. (2023a). Math-Dial: A Dialogue Tutoring Dataset with Rich Pedagogical Properties Grounded in Math Reasoning Problems. *Findings of EMNLP 2023*. <https://doi.org/10.18653/v1/2023.findings-emnlp.372>
- Mayer, R. E., & Moreno, R. (2003). Nine Ways to Reduce Cognitive Load in Multimedia Learning. *Educational Psychologist*, 38(1), 43-52. https://doi.org/10.1207/S15326985EP3801_6
- Miller, P., & DiCerbo, K. (2024). LLM Based Math Tutoring: Challenges and Dataset. EdArXiv. <https://doi.org/10.35542/osf.io/5zwv3>
- Nickow, A., Oreopoulos, P., & Quan, V. (2024). The Promise of Tutoring for PreK-12 Learning: A Systematic Review and Meta-Analysis of the Experimental Evidence. *American Educational Research Journal*, 61(1), 74-107. <https://doi.org/10.3102/00028312231208687>
- Renkl, A. (2014). Toward an Instructionally Oriented Theory of Example-Based Learning. *Cognitive Science*, 38(1), 1-37. <https://doi.org/10.1111/cogs.12086>
- Roediger, H. L., & Karpicke, J. D. (2006). Test-Enhanced Learning: Taking Memory Tests Improves Long-Term Retention. *Psychological Science*, 17(3), 249-255. <https://doi.org/10.1111/j.1467-9280.2006.01693.x>
- Rohrer, D., & Taylor, K. (2007). The shuffling of mathematics problems improves learning. *Instructional Science*, 35(6), 481-498. <https://doi.org/10.1007/s11251-007-9015-8>

- Shute, V. J. (2008). Focus on Formative Feedback. *Review of Educational Research*, 78(1), 153-189. <https://doi.org/10.3102/0034654307313795>
- Sweller, J., & Cooper, G. A. (1985). The Use of Worked Examples as a Substitute for Problem Solving in Learning Algebra. *Cognition and Instruction*, 2(1), 59-89. https://doi.org/10.1207/s1532690xi0201_3
- VanLehn, K. (2011). The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems. *Educational Psychologist*, 46(4), 197-221. <https://doi.org/10.1080/00461520.2011.611369>
- Vygotsky, L. S. (1978). *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press (book). <https://www.hup.harvard.edu/books/9780674576292>
- Wood, D., Bruner, J. S., & Ross, G. (1976). The Role of Tutoring in Problem Solving. *Journal of Child Psychology and Psychiatry*, 17(2), 89-100. <https://doi.org/10.1111/j.1469-7610.1976.tb00381.x>

About this paper: written by the DuckyHelper team at MingLLM, Inc. It describes the design of DuckyHelper and cites published research. It has not been peer reviewed and reports no study of DuckyHelper. HTML version: <https://dukyhelper.com/research/papers/built-to-teach/>